

---

# Parallel Computational Acoustics Library

## User's Guide

by F. Magoulès and F.-X. Roux

---

June 20, 2002

### Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>The Mesh Files Format</b>	<b>2</b>
<b>3</b>	<b>The Mesh Generation File</b>	<b>3</b>
<b>4</b>	<b>The Mesh Definition File</b>	<b>5</b>
<b>5</b>	<b>The Domain Decomposition File</b>	<b>8</b>
<b>6</b>	<b>The Model Property File</b>	<b>9</b>
<b>7</b>	<b>The WAV File</b>	<b>13</b>
<b>8</b>	<b>The VTK File</b>	<b>15</b>

## 1 Introduction

The PCA-Lib User's Guide v1.0 is the manual for the PCA-Lib functions library and defines the PCA-Lib data file format. The library comprises three main types of function - those for pre-processing (mesh data), those for processing (involving numerical and matrix analysis), and those for post-processing (visualization, noise rendering).

This work was motivated by the need to integrate the various finite element, mesh generation, mesh partitioning, domain decomposition methods and parallel solvers, and plotting programs developed by the group. At present, the PCA-Lib contains the most recent and powerful development in finite element methods for acoustics and parallel iterative domain decomposition methods. There are two groups of algorithms in the library: the first group of algorithms is based on well established methods which are generally used in the industry while the second group of algorithms use the current result of research. This helps the library to be use at the same time by industrial partners and academics researchers. The library is therefore able to solve huge acoustic problems which were unable to solve still now.

## 2 The Mesh Files Format

The PCA-Lib format supports meshes of a single element type and mixed meshes. The PCA-Lib mesh file format takes the form of a series of keywords followed by one or more data items. For example,

```
FREQ
  1
  1.273239544735163
```

denotes that one frequency is considered for the acoustic problem and that the value of this frequency is equal to 1.273239544735163 Hertz.

There are currently four types of data file – the mesh generation (**.gen**), the mesh definition (**.smsh**), the model property (**.nson**) and the domain decomposition (**.nsplit**). The mesh generation file describe the mesh parameters and the partitioning characteristics of a regular domain. The mesh definition file contains the geometric and topological description of the mesh. The domain decomposition file describes the partitioning of the domain into subdomains. The model property file contains all the characteristics of the problem. The following sections describe the keywords and the keyword data for the above mentioned data files. Each section contains a table which lists the keywords in the order they must be declared and defines the associated keyword data. The data type – whether it is an integer (i), real (r) or a character string (s) – is also given, where, for example, 'i 3\*r' denotes that the data consist of an integer followed by three reals. Units, where relevant, are enclosed in square brackets ([. .]).

**IMPORTANT** Keyword data must follow the keyword on the next line. Most keywords are optional, but those which are declared must be follow by the associated keyword data in a strict, predefined order. This order automatically accounts for dependencies between the data. If the number of data is zero then the keyword could be omitted.

### 3 The Mesh Generation File

The mesh generation file (**.gen**) contains all the data required by the PCA-Lib mesh generator. The geometry of the domain is specified, as well as the mesh, domain decomposition properties and boundary conditions.

The domain is a block domain mesh with regular hexahedron elements. The boundary conditions are set to the value of a three dimensional planar wave. The planar wave is expressed by the following convention  $e^{i\vec{k}\vec{x}}$  where  $\vec{k}$  represents the propagation vector of the wave and  $\vec{x}$  the coordinates of the current point

$$\vec{k} = \begin{pmatrix} \omega \cos(\theta)\cos(\varphi) \\ \omega \sin(\theta)\cos(\varphi) \\ \omega \sin(\varphi) \end{pmatrix} \quad \vec{x} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \quad (1)$$

with the wavenumber  $\omega$  and the polar coordinates  $\theta$  and  $\varphi$ . It is possible to impose Dirichlet, Neumann or Robin boundary conditions on each side of the domain through the specification of boolean flags on the front, back, right, left, top and bottom faces of the domain.

**IMPORTANT** For the mesh generation file, each keyword data must be written on a different line and can be followed by comments.

Keyword	Keyword Data	Data Type
MESH GENERATION	size of domain in $x$ direction	r
	size of domain in $y$ direction	r
	size of domain in $z$ direction	r
	number of cells per subdomain in $x$ direction	i
	number of cells per subdomain in $y$ direction	i
	number of cells per subdomain in $z$ direction	i
	number of subdomains in $x$ direction	i
	number of subdomains in $y$ direction	i
	number of subdomains in $z$ direction	i
	Dirichlet boundary conditions on front: $x > 0$	i
	Dirichlet boundary conditions on back: $x < 0$	i
	Dirichlet boundary conditions on right: $y > 0$	i
	Dirichlet boundary conditions on left: $y < 0$	i
	Dirichlet boundary conditions on top: $z > 0$	i
	Dirichlet boundary conditions on bottom: $z < 0$	i
	Robin boundary conditions on front: $x > 0$	i
	Robin boundary conditions on back: $x < 0$	i
	Robin boundary conditions on right: $y > 0$	i
	Robin boundary conditions on left: $y < 0$	i
	Robin boundary conditions on top: $z > 0$	i
	Robin boundary conditions on bottom: $z < 0$	i
	Neumann boundary conditions on front: $x > 0$	i
	Neumann boundary conditions on back: $x < 0$	i
	Neumann boundary conditions on right: $y > 0$	i
	Neumann boundary conditions on left: $y < 0$	i
	Neumann boundary conditions on top: $z > 0$	i
	Neumann boundary conditions on bottom: $z < 0$	i
	frequency [Hertz]	r
	$\theta$ polar coordinates [radian]	r
	$\varphi$ polar coordinates [radian]	r

Table 1: The mesh generation file keywords and keyword data

An example **.gen** mesh generation file with comments would be the following.

```

TITLE MESH GENERATION
1.2 , size of domain in x direction
0.6 , ----- in y direction
0.3 , ----- in z direction
40 , number of cells per subdomain in x direction
20 , ----- in y direction
10 , ----- in z direction
1 , number of subdomains in x direction
1 , ----- in y direction
1 , ----- in z direction
1 , Dirichlet boundary conditions on front : x > 0
1 , ----- on back : x < 0
0 , ----- on right : y > 0
0 , ----- on left : y < 0
0 , ----- on top : z > 0
0 , ----- on bottom : z < 0
0 , Robin boundary conditions on front : x > 0
0 , ----- on back : x < 0
1 , ----- on right : y > 0
1 , ----- on left : y < 0
1 , ----- on top : z > 0
1 , ----- on bottom : z < 0
0 , Neumann boundary conditions on front : x > 0
0 , ----- on back : x < 0
1 , ----- on right : y > 0
1 , ----- on left : y < 0
1 , ----- on top : z > 0
1 , ----- on bottom : z < 0
1.273239544735163 , frequency
0.39269908169872414 , theta-angle
0.39269908169872414 , phi-angle

```

An example mesh generation file

The geometry of the associated domain and the mesh are shown in Figures 3 and 4 and the acoustic solution that will be obtained after an acoustic simulation is shown in Figures 5 and 6.

## 4 The Mesh Definition File

The mesh definition file (**.smsh**) contains the main description of the mesh. At present two different elements types are supported – one quadrilateral element (**QUAD1**) and one block element (**BLOCK1**). These element types are defined in Figure 1 which shows the order in which the element nodes must be labelled and Table 2 gives a short description as well. Individual elements have uniform material properties.

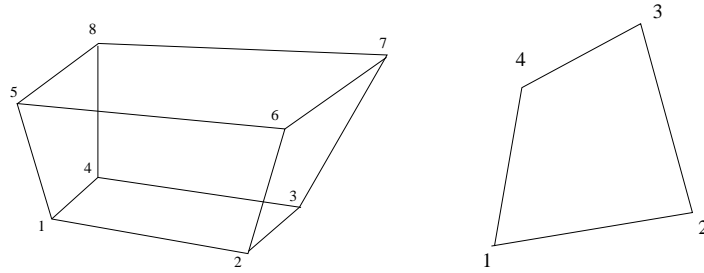


Figure 1: The element node orders.

Element Type	Mesh Type	Description	Number of Vertices	Number of Nodes
<b>QUAD1</b>	10	Surface element	4	4
<b>BLOCK1</b>	1	Volume element	8	8

Table 2: The element types

An example mesh definition file is shown in the following, and the result can be seen in Figure 2.

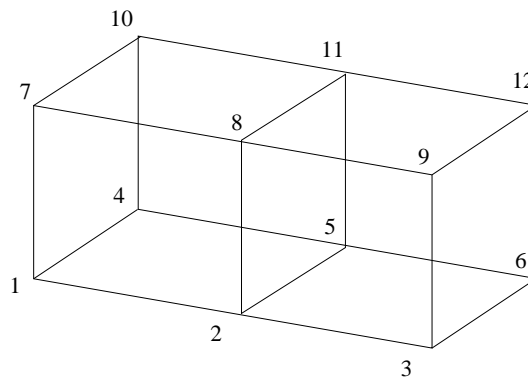


Figure 2: An example two blocks mesh.

```

NODES
12
1 0.0E+00 0.0E+00 0.0E+00
2 0.5E+00 0.0E+00 0.0E+00
3 0.1E+01 0.0E+00 0.0E+00
4 0.0E+00 0.1E+01 0.0E+00
5 0.5E+00 0.1E+01 0.0E+00
6 0.1E+01 0.1E+01 0.0E+00
7 0.0E+00 0.0E+00 0.1E+01
8 0.5E+00 0.0E+00 0.1E+01
9 0.1E+01 0.0E+00 0.1E+01
10 0.0E+00 0.1E+01 0.1E+01
11 0.5E+00 0.1E+01 0.1E+01
12 0.1E+01 0.1E+01 0.1E+01
FEM
2
1 1 1 1 2 5 4 7 8 11 10
2 1 1 2 3 6 5 8 9 12 11
FEM
1
1 2 10 3 6 12 9

```

An example mesh definition file

A mesh is defined in terms of a set of *mesh-points* which are joined by straight lines to form the edges or surfaces of the elements.

The **NODES** keyword must be followed by the number of mesh-points and the nodes coordinates.

The keyword **FEM** declares the type of elements in the mesh and it also provides the number of elements as well as the geometry. The elements can be volume or surface elements. The elements are defined in the sense that, in the variational formulation, the integration of the shape functions of the elements is directly added to the admittance matrix of the problem, e.g. the left hand side. For this reason, an element can be either a volume element or a surface element (when Robin boundary conditions are applied on it). Indeed, after the variational formulation the Robin boundary condition is added to the left hand side and to the right hand side. In the previous case, the mesh is composed of two **BLOCK1** elements and one **QUAD1** element. This means that a homogeneous Robin boundary condition is applied on the front of the domain.

Another example is presented to show the use of extra keywords. More precisely, the **FAC** keyword is used in order to define the right hand side value of the previous Robin boundary condition.

Keyword	Keyword Data	Data Type
<b>NODES</b>	Number of nodes	i
	For each node, the node id followed, by the three coordinates of the node	i 3*r
<b>FEM</b>	Number of elements	i
	For each element, the element id followed by the acoustic region where the element belongs, the element type and for each vertex (e.g. node) of the element, the node id	3*i *i
<b>FAC</b>	Number of clamped facet	i
	For each clamped facet, the clamped facet id followed by the admittance region where the facet belongs, the facet (e.g. element) type and for each vertex (e.g. node) of the facet, the node id and for each vertex (e.g. node) of the facet, the real and imaginary value of the imposed velocities	3*i *i *r

Table 2: The mesh definition file keywords and keyword data

```

NODES
12
1 0.0E+00 0.0E+00 0.0E+00
2 0.5E+00 0.0E+00 0.0E+00
3 0.1E+01 0.0E+00 0.0E+00
4 0.0E+00 0.1E+01 0.0E+00
5 0.5E+00 0.1E+01 0.0E+00
6 0.1E+01 0.1E+01 0.0E+00
7 0.0E+00 0.0E+00 0.1E+01
8 0.5E+00 0.0E+00 0.1E+01
9 0.1E+01 0.0E+00 0.1E+01
10 0.0E+00 0.1E+01 0.1E+01
11 0.5E+00 0.1E+01 0.1E+01
12 0.1E+01 0.1E+01 0.1E+01
FEM
2
1 1 1 1 2 5 4 7 8 11 10
2 1 1 2 3 6 5 8 9 12 11
FEM
1
1 2 10 3 6 12 9
FAC
1
1 2 4 3 6 12 9 0.1E+02 -0.6E+01 ... -0.6E+01

```

An example mesh definition file

An exhaustive list of possible keywords in an **.smsh** file is presented in Table 2.

## 5 The Domain Decomposition File

The domain decomposition file (**.nsplit**) specifies the subdomains into which the elements have been partitioned (Table 3). The indices must lie between one and the number of subdomains inclusive. The total number of elements listed in the file must be consistent with the total number of elements defined in the mesh definition file, in the sense that both volume and surface elements (if there are) must be listed. In this case, the volume elements are listed first and then the surface elements.

Keyword Data	Data Type
Number of elements	i
Subdomain index of each <b>BLOCK1</b> element	i
Subdomain index of each <b>QUAD1</b> element	i

Table 3: The domain decomposition file keywords and keyword data

An example domain decomposition file is shown for the mesh presented in Figure 2.

```
3 , number of elements including facet
1
2
2
```

An example domain decomposition file



## 6 The Model Property File

The model property file (**.nson**) contains all the characteristics of the problem including for example the frequencies, acoustic and admittance material properties, loading curves, multiple constraints points, nodal pressure or velocity, which are declared in the mesh definition file. The different sections of the model file are separated with keywords listed in Tables 4 and 5. The different keywords can be defined in any order.

The definition of acoustic material properties follows the keyword **ACOU** which refers to the acoustic region, e.g. to some volume elements. The definition of the admittance material properties follow the keyword **ADMI** which refers to the admittance region, e.g. to some surface elements where Robin type boundary conditions are applied. Any acoustic material and admittance material properties can be defined and not just those used by the current mesh. This enables a number of different meshes to use a single **ACOU** and **ADMI** section.

The Dirichlet boundary conditions can be imposed using the keyword **NPRE**, which sets the acoustic nodal pressure at some nodes.

The definition of the frequencies begins with the keyword **FREQ**, and contains the total number of frequencies and for each frequency the magnitude of it.

The parameter definitions for the solver begin with the keyword **SOLV**. It is important to notice that all the parameters associated with the solver are written on the same line.

The mesh file name follows the keyword **MESH FILE** and refers to a mesh definition file (**.smsh**).

For the use of keywords not discussed here see the relevant section in the documentation.

Keyword	Keyword Data	Data Type
<b>TITLE</b>	Title of model property file	s
<b>ACOU</b>	Number of acoustic region For each acoustic region: - the acoustic material id - the acoustic material type - the density - the curve density id - the real part of the celerity - the imaginary part of the celerity - the curve celerity id	i 4*i 2*r i
<b>ADMI</b>	Number of admittance region For each admittance region: - the admittance material id - the admittance material type - the real part of the <b>impedance</b> - the imaginary part of the <b>impedance</b> - the curve <b>impedance</b> id	i 2*i 2*r i
<b>ELLIPSE</b>		
<b>NPRE</b>	List id of clamped nodes followed by the number of clamped nodes For each clamped node: - the node id - the type of degree of freedom - the real part of the imposed nodal pressure - the imaginary part of the imposed nodal pressure - the id curve	2*i 2*i 2*r i
<b>NVEL</b>		
<b>FREQ</b>	Number of frequencies For each frequency, the magnitude [Hertz]	i *r
<b>CURV</b>		
<b>MPC</b>		

---

Table 4: The model property keywords and keyword data

Keyword	Keyword Data	Data Type
SOLV	<b>type_galerkin</b>	0 (= Petrov-Galerkin) 1 (= Galerkin Least Square) 2 (= Galerkin Gradient Least Square) i
	<b>symmetric_matrix</b>	0 (= unsymmetric admittance matrix) 1 (= symmetric admittance matrix) i
	<b>type_solver</b>	1 (= direct solver) 2 (= parallel orthodir iterative method) 3 (= FETI 1 field) 4 (= FETI 2 fields) i
	<b>type_precond</b>	0 (= no preconditionner) 1 (= diagonal scaling if type_solver equal 2) 1 (= lumped if type_solver equal 3) 2 (= incomplete LU if type_solver equal 2) 2 (= Dirichlet if type_solver equal 3) i
	<b>max_numb_it</b>	— (= maximum number of iterations) i
	<b>numb_dir</b>	— (= number of stored directions (restart)) i
	<b>out_of_core</b>	0 (= in core direct solver) 1 (= out of core direct solver) i
	<b>epsilon</b>	— (= stopping criterion for all iterative solvers) r
	<b>print_matrix</b>	0 (= no printing for sparse admittance matrix) 1 (= print sparse admittance matrix of subdomain) i
	<b>print_rhs</b>	— (= no effect) i
	<b>print_solution</b>	0 (= no printing for solution) 1 (= print local solution in free format) 2 (= print local solution in FEMVIEW format) 3 (= print local solution in VTK format) i
	<b>print_mesh</b>	0 (= no printing for mesh) 1 (= print global mesh in FEMVIEW format) 2 (= print local mesh in FEMVIEW format) i
	<b>type_opt</b>	0 (= standard regularization matrix) 1 (= optimal regularization matrix) 2 (= sparse approximation of optimal regularization matrix) i
	<b>cache_size</b>	— (= size of primary cache memory in Kbytes) i
	<b>nsd</b>	— (= no effect) i
MESH FILE	<b>file_name</b>	Mesh definition filename s

Table 5: The model property keywords and keyword data (cont.)

An example **.nson** file with one acoustic region and one admittance region would be the following.

```

GENERATED BY SPLIT_MESH (PCA-LIB V1.0)
ACOU
1
1 1 0.1E+01 1 0.1E+01 0.0E+00 1
ADMIT
1
2 1 0.1E+01 0.0E+00 1 outlet
FREQ
1
1.273239544735163
SOLV Defaults
0 1 3 1 200 200 0 0.1E-05 0 0 3 0 1 256 12
MESH FILE
mesh.smesh

```

An example model property file

While the corresponding **.smsh** file looks like this.

```

NODES
12
1 0.0E+00 0.0E+00 0.0E+00
2 0.5E+00 0.0E+00 0.0E+00
3 0.1E+01 0.0E+00 0.0E+00
4 0.0E+00 0.1E+01 0.0E+00
5 0.5E+00 0.1E+01 0.0E+00
6 0.1E+01 0.1E+01 0.0E+00
7 0.0E+00 0.0E+00 0.1E+01
8 0.5E+00 0.0E+00 0.1E+01
9 0.1E+01 0.0E+00 0.1E+01
10 0.0E+00 0.1E+01 0.1E+01
11 0.5E+00 0.1E+01 0.1E+01
12 0.1E+01 0.1E+01 0.1E+01
FEM
2
1 1 1 1 2 5 4 7 8 11 10
2 1 1 2 3 6 5 8 9 12 11
FEM
1
1 2 10 3 6 12 9

```

An example mesh definition file

## 7 The WAV File

The audio formatted file (**.wav**) contains the digital signal as stored in a WAV file format. The results of this format can be listened in the loudspeakers of all computer using a Microsoft Windows system or an equivalent emulator. This file can be open through a hexadecimal editor, but the description of the format is rather difficult to understand without a full analysis.

The WAV files are probably the simplest of the common formats for storing audio samples. Contrary to other type of files which are compressed, files WAV store the data "in the raw", without any other operation than the formatting of these data. A WAV file is divided into three principal blocks of information: the RIFF block which identifies the file like a WAV file; the FORMAT block which identifies the parameters of the file as sample rates; the DATA block which contains the data themselves. Each block is composed of a series of Bytes (1 Byte = 8 Bits, 1 Bit = a primary data) formatted according to methodology indicated Table 6, 7, 8.

Byte Number	Data Stored	Description	Formatting
0 - 3	"RIFF"	Beginning of RIFF Block	ASCII Characters
4 - 7	—	Total length of package (file) to follow	Binary 'little endian'
8 - 11	"WAVE"	Identify the file as a WAV file	ASCII Characters

Table 6: RIFF Block (total length 12 Bytes)

Byte Number	Data Stored	Description	Formatting
0 - 3	"fmt "	Beginning of FORMAT Block	ASCII Characters
4 - 7	0x10	Length of FORMAT Block	Binary 'little endian'
8 - 9	0x01	Microsoft format	Binary 'little endian'
10 - 11	0x01 (= mono) 0x02 (= stereo)	Number of channels	Binary 'little endian'
12 - 15	—	Number of sample per second (Hz)	Binary 'little endian'
16 - 19	—	Number of Bytes per second	Binary 'little endian'
20 - 21	1 (= 8 Bit mono) 2 (= 8 Bit stereo) 2 (= 16 Bit mono) 4 (= 16 Bit stereo)	Number of Bytes per samples	Binary 'little endian'
22 - 23	—	Number of Bits per samples	Binary 'little endian'

Table 7: FORMAT Block (total length 24 Bytes)

An example of a WAV file is presented bellow. The displaying of the file by a hexadecimal editor shows the Byte values in the files.

```
52 49 46 46 D0 3B 01 00 57 41 56 45 66 6D 74 20
10 00 00 00 01 00 02 00 22 56 00 00 88 58 01 00
04 00 10 00 64 61 74 61 7C 3B 01 00 F8 FF 01 00
FF FF FE FF 05 00 02 00 05 00 03 00 03 00 FD FF
...
```

Byte Number	Data Stored	Description	Formatting
0 - 3	"data"	Beginning of DATA Block	ASCII Characters
4 - 7	—	Length of data to follow	Binary 'little endian'
8 - end	—	Data samples	Binary 'little endian'

Table 8: DATA Block

An example WAV file

• **Line 1 (total length 16 Bytes)**

As indicated in the previous table, the file begins with the **four Bytes** corresponding to ASCII Characters "RIFF". **Four Bytes** follows it which indicate the total length in Bytes of the continuation of the file (integer value 80848), e.g. the size of the file on the disc (integer value 80856) minus the size of word "RIFF" (integer value 8). The next **eight Bytes** correspond to ASCII Characters "WAVE", and to the ASCII Characters "fmt".

• **Line 2 (total length 16 Bytes)**

The first **four Bytes** represent the length of the block FORMAT (integer value 16), followed by **two Bytes** indicating that it is a Microsoft file (integer value 1). The next **two Bytes** show that it is about a stereo recording (integer value 2). The next **four Bytes** indicate the number of samples per second (integer value 22050). The next **four Bytes** indicate the number of Bytes per second (integer value 88200) since we have  $(88200 = 22050 * 4)$ , where 4 represents the number of Bytes per sample.

• **Line 3 - end**

The next **two Bytes** represent the number of Bytes per sample (integer value 4), since we have a 16 Bits stereo recording. The next **two Bytes** represent the number of Bits per sample, (integer value 16) and the next **four Bytes** correspond to the ASCII Characters "data". Then comes **four Bytes** indicating the number of Bytes of the data which follow (integer value 80764). The **following Bytes** which extend until the end of the file correspond to the coding of the samples.

## 8 The VTK File

The Visualization ToolKit file format (**.vtk**) contains the acoustic level as stored in a VTK file format. The results of this format can be visualized on all computer since the VTK library is used. The VTK file display the acoustic level on a unstructured mesh. The keywords and keywords datas are listed in the following table.

Keyword	Keyword Data	Data Type
<b>POINTS</b>	Number of points followed by the data type of the coordinates here <b>float</b> . For each node, the three coordinates of the node.	i s 3*r
<b>CELLS</b>	Number of cells followed by the total number of entry to read (for internal use inside VTK). For each cell, the number of nodes of the cell followed by the nodes id of the cell	i i i *i
<b>CELL_TYPES</b>	Number of cells, followed by the list of cells type (12 = hexahedrom, 9 = quad)	i *i
<b>POINT_DATA</b>	Number of data, followed by the data type here <b>scalars</b> . For each node the associated value.	i *s *r

Table 9: The VTK keywords and keyword data

An basic example of such a file is now presented.

```

# vtk DataFile Version 2.0
3D scalar data
ASCII

DATASET UNSTRUCTURED_GRID
POINTS 12 float
 0.0E+00 0.0E+00 0.0E+00
 0.5E+00 0.0E+00 0.0E+00
 0.1E+01 0.0E+00 0.0E+00
 0.0E+00 0.1E+01 0.0E+00
 0.5E+00 0.1E+01 0.0E+00
 0.1E+01 0.1E+01 0.0E+00
 0.0E+00 0.0E+00 0.1E+01
 0.5E+00 0.0E+00 0.1E+01
 0.1E+01 0.0E+00 0.1E+01
 0.0E+00 0.1E+01 0.1E+01
 0.5E+00 0.1E+01 0.1E+01
 0.1E+01 0.1E+01 0.1E+01

CELLS 3 23
 8 0 1 4 3 6 7 10 9
 8 1 2 5 4 7 8 11 10
 4 0 6 9 3

CELL_TYPES 3
12
12
9

POINT_DATA 12
SCALARS scalars float 1
LOOKUP_TABLE default
-0.28505
 0.18216
-0.14550
-0.28505
 0.18216
-0.14550
-0.28505
 0.18216
-0.14550
-0.28505
 0.18216
-0.14550

```

An example VTK file



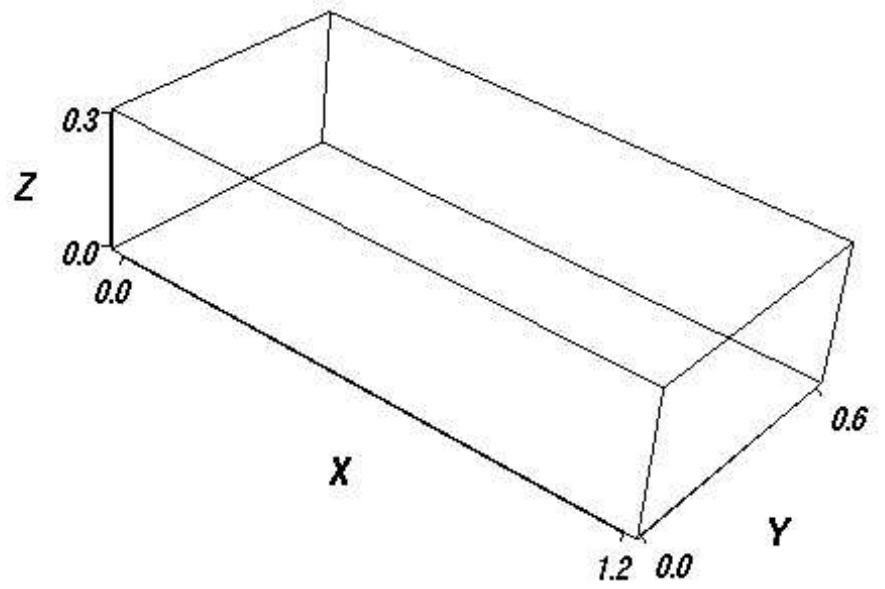


Figure 3: An example of a three dimensional domain.

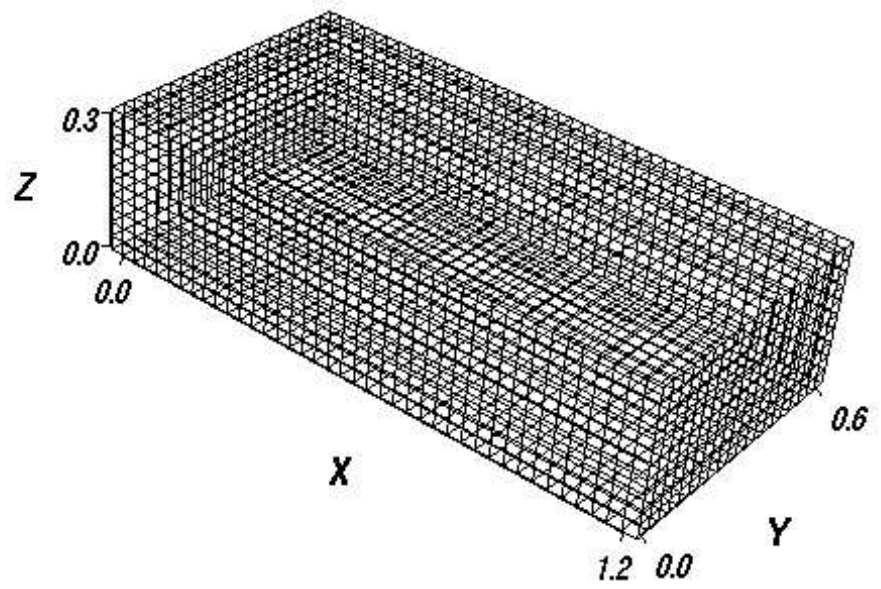


Figure 4: An example of a regular three dimensional mesh.

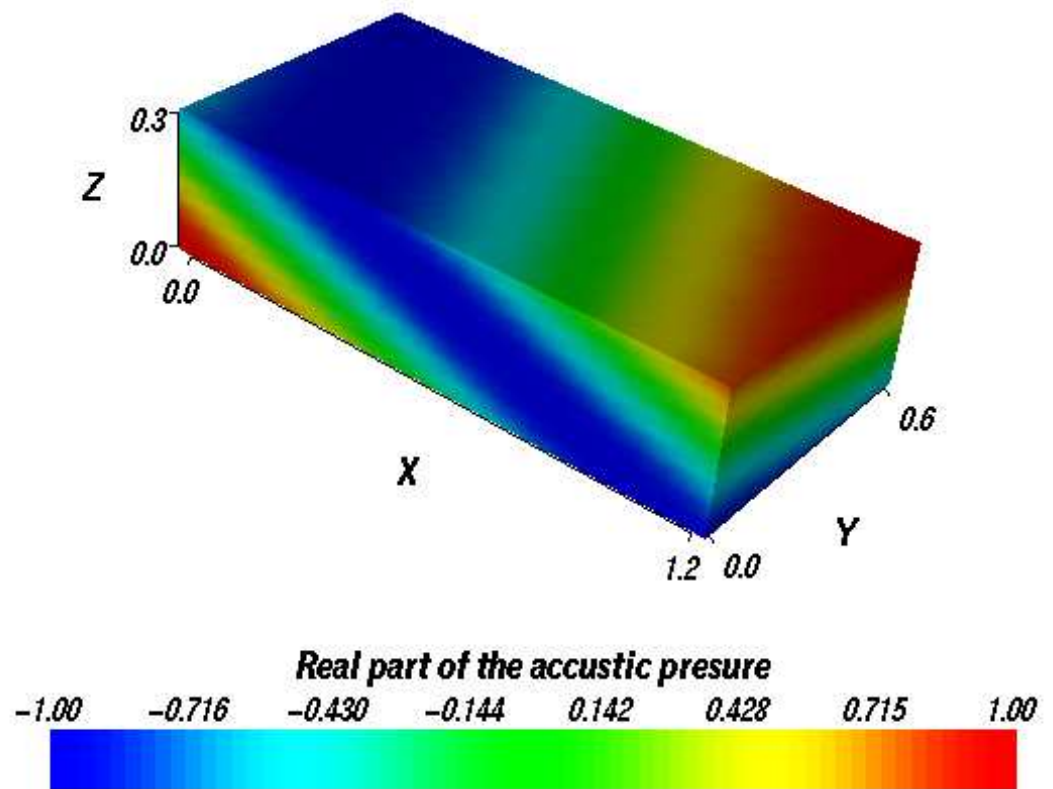


Figure 5: An example of a three dimensional planar wave ( $\theta = \pi/8$ ,  $\varphi = \pi/8$ ).

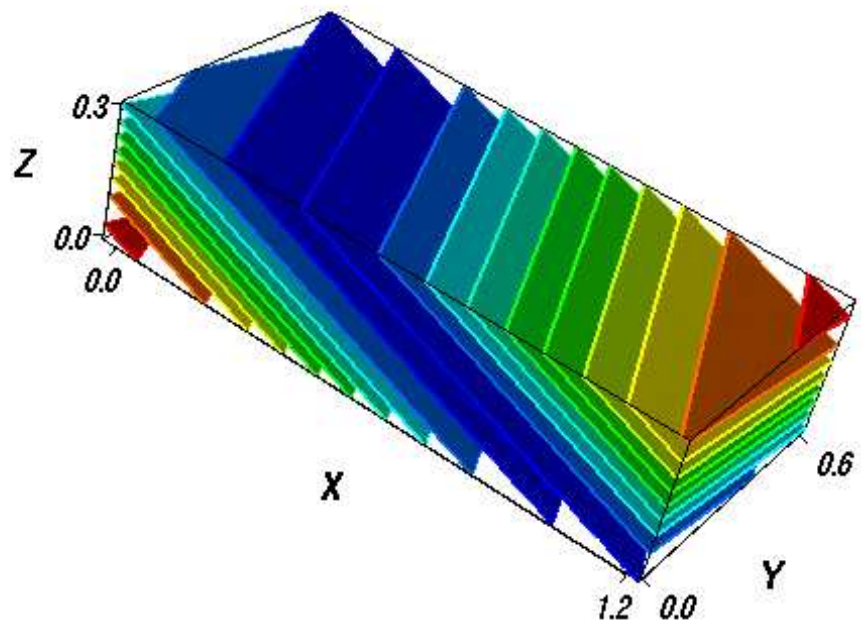


Figure 6: An example of a three dimensional planar wave ( $\theta = \pi/8$ ,  $\varphi = \pi/8$ ). Isolines of the real part of the acoustic pressure.

For further information please see, the “**PCA-Lib Reference Manual**”.

## **Acknowledgements**

The authors acknowledge partial financial support by the European Community under the Enhancing Access to Research Infrastructure action of the Improving Human Potential programme, contract number HPRI-1999-CT-0026.